

(19)  Canadian Intellectual Property Office
An Agency of Industry Canada

Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

(11) CA 2 279 145 (13) A1

(43) 01.02.2000

(12)

(21) 2 279 145

(51) Int. Cl.⁶:

G06F 017/30, H04N 007/01

(22) 30.07.1999

(30) 09/127,613 US 01.08.1998

(71) COREL CORPORATION,
1600 Carling Avenue, OTTAWA, O1 (CA).

(72) CALL, JUSTIN D. (US).

(74) SIM & MCBURNEY

(54) HYPERLIEN A VISIONNEMENT PREALABLE DU DOCUMENT

(54) HYPERTEXT LINK PREVIEW

(57)

In a computer application for displaying hypertext documents, a system and method for displaying a preview of a linked document before the link is activated by the user. The displayed preview is generated from the linked document itself, in order to provide an accurate and timely representation of the linked document.

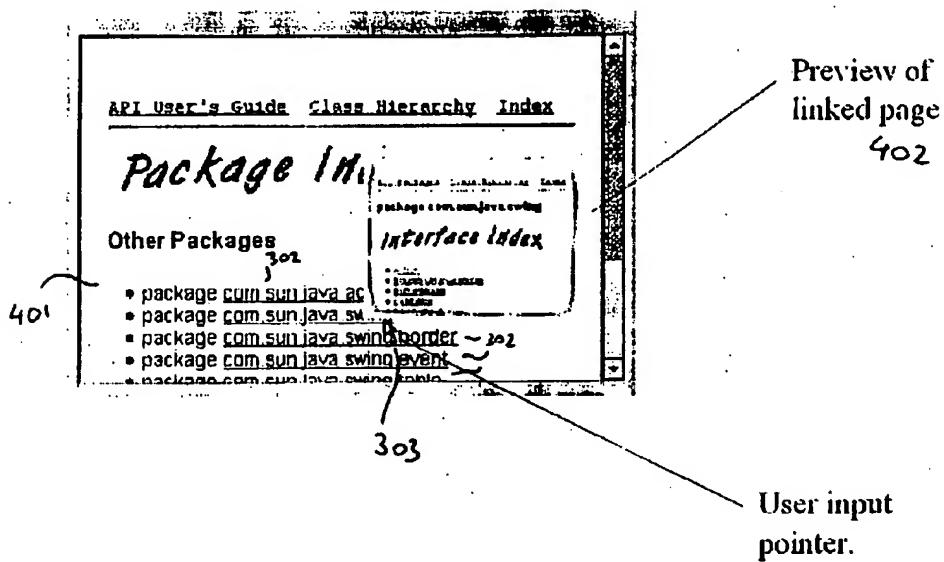


(21)(A1) **2,279,145**

(22) 1999/07/30

(43) 2000/02/01

(72) CALL, JUSTIN D., US
(71) COREL CORPORATION, CA
(51) Int.Cl.⁶ G06F 17/30, H04N 7/01
(30) 1998/08/01 (09/127,613) US
(54) **HYPERLIEN A VISIONNEMENT PREALABLE DU DOCUMENT**
(54) **HYPertext Link Preview**



(57) In a computer application for displaying hypertext documents, a system and method for displaying a preview of a linked document before the link is activated by the user. The displayed preview is generated from the linked document itself, in order to provide an accurate and timely representation of the linked document.



HYPertext LINK PREVIEW

ABSTRACT OF THE DISCLOSURE

In a computer application for displaying hypertext documents, a system and method for displaying a preview of a linked document before the link is activated by the user. The displayed preview is generated from the linked document itself, in order to provide an accurate and timely representation of the linked document.

HYPertext Link PREVIEW

Inventor:
Justin D. Call

BACKGROUND OF THE INVENTION

5

1. Field of the Invention

The present invention relates generally to software applications such as browsers for viewing hypertext documents, and more particularly to a system and method of displaying a preview of a linked document before the link is activated.

10 **2. Description of Background Art**

Hypertext is a form of text, usually displayed on a computer screen, which facilitates linking from one document to another. In recent years, standardized formats for hypertext encoding have emerged, such as for example Hypertext Markup Language (HTML), Extended Markup Language (XML), Standardized General Markup Language (SGML), and the like. In general, such hypertext languages provide codes for specifying links from one document to another. Links are usually associated with a particular word, phrase, graphic, or portion of a graphic, in a document. Each such link specifies a document that is to be retrieved when the link is activated. Many links, often leading to different documents, may be provided in a single document.

20 Currently, the most common use for hypertext is as a medium for navigating the World Wide Web and similar computer networks. By specifying links on a document (or page), the document author is able to provide a mechanism for permitting the user to retrieve associated documents from various web sites, as well as navigate among documents within a site.

25 Users view and navigate hypertext documents using software such as a browser. Several popular browsers are available, including for example, Netscape® Navigator™ from Netscape Communications, and Microsoft® Internet Explorer™ from Microsoft Corporation. Operation of such browsers is generally quite similar.

Documents encoded in HTML or a similar language are displayed on a computer screen as "pages", formatted according to specified codes embedded in the document, and often including graphical elements. Links to other documents may also be present on a displayed document, and the text associated with a link is often

5 indicated by the use of a distinct font style and/or color. In some cases, a graphic element or a portion thereof may be associated with a link. In order to activate a link on a displayed document, the user typically moves a mouse or similar pointing device so as to move an on-screen cursor onto the text or graphic associated with the link, and clicks a button. This causes the document associated with the selected link

10 to be retrieved.

In most conventional browsers, the user is not able to view a linked document until he or she activates the link. In many situations, the user may not be sure whether a particular link will retrieve a document that is of interest. Generally, hypertext documents provide some clues as to the nature of the linked document,

15 giving the user some guidance in deciding which link to activate next. The particular text or graphic associated with a link is often the best source of such information.

However, the information provided in the first, or source, document itself is often insufficient in describing the linked document. Browsers sometimes provide features for improving the information available to the user before clicking on a link.

20 For example, both Navigator and Internet Explorer momentarily display the file name of a linked document whenever the user positions the on-screen cursor over the link. However, in many situations the file name is obscure and/or meaningless, since it is essentially an arbitrary name selected by the document author.

Some browsers also provide functionality for displaying a "ToolTip" when the on-screen cursor is placed on top of a link. This ToolTip displays a short line of text, which presumably can help describe the link. However, the text displayed in the ToolTip is specified by the document author, as an ALT tag in the HTML code specifying the link parameters; if the author does not provide such a tag, or provides inadequate text for the tag, the ToolTip display may not be helpful to the user.

Some web pages include more sophisticated methods of describing links, using scripting environments such as JavaScript or similar. Using such techniques, a graphic and/or short description of a link may be presented to the user when the cursor is placed over a link. However, such techniques still employ information from the script presented in the source document, and do not use information from the linked document. Therefore the descriptions presented are not automatically generated or updated based on the content of the linked document. Furthermore, such techniques rely on the source document's author to provide the necessary programming and data to present the link previews. Thus, they are poor solutions for automated generation and update of usable and detailed link previews.

In general, all the above techniques rely on information in the source document, and do not display any information taken from the linked document itself. Where the source document does not provide adequate information, there is no way, using the above techniques, for the user to be presented with sufficient "preview" information to decide whether to follow a link. Furthermore, any information taken from the source document can potentially become outdated, as the linked document may change without the source document being updated accordingly.

What is needed is a system and method of displaying an up-to-date preview of a linked document that provides sufficient information regarding the linked document so that a user can decide whether or not to activate a link. What is further needed is a preview system and method that does not rely on a document author to provide information about a linked document. What is further needed is a preview system and method that obtains preview information from the linked document. Finally, what is further needed is a preview system and method that reflects updated information concerning the linked document, without manual update of the source document.

SUMMARY OF THE INVENTION

In accordance with the present invention, there is provided a system and method of displaying a preview of a linked document, in response to a user showing some interest in the document. For example, an image resembling either a miniature 5 version of the linked document, or a portion of the linked document, can be displayed when the user passes an on-screen cursor over the link. By providing a graphic representation of the linked document, the present invention is able to provide more information than conventional methods which display only a limited text-based description of a link that is based solely on information from the source 10 document.

In the present invention, the displayed preview image is based on the linked document itself, rather than on any hard-coded information from the source document. Thus, the displayed preview is up-to-date even if the source document has not been changed to reflect the changes in the linked document.

15 In one embodiment, preview data for a link is retrieved and cached in advance using a background process, so that previews can be displayed without undue delay when the user places the cursor over a link. Link preview resources for displaying previews for all links on a source document begin to load as soon as the source document has been loaded. When the user indicates interest in a particular link, the 20 corresponding preview can immediately be displayed if its resources have been loaded. If the loading has not yet been performed or completed when a user expresses interest in a link, the present invention initiates immediate loading of the resources needed for that link. When such loading is complete, the preview is displayed for the user. Display of the preview may be stopped when appropriate, for 25 example when a predefined period of time has elapsed, or when the user indicates that he or she is no longer interested in the link.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram of a system for practicing the present invention.

Fig. 2 is an example of a hypertext document containing links to other documents.

5 Fig. 3 is an example of a screen display showing a hypertext document according to the prior art.

Fig. 4 is an example of a screen display showing a hypertext document and a link preview according to the present invention.

10 Fig. 5 is a block diagram of the functional components of one embodiment of the present invention.

Fig. 6A is a timeline diagram showing a page loading operation according to the present invention.

Fig. 6B is a flowchart showing a page loading operation according to the present invention.

15 Fig. 7A is a timeline diagram showing a link previewing operation according to the present invention.

Fig. 7B is a flowchart showing a link previewing operation according to the present invention.

20 Fig. 8A is a timeline diagram showing a preview stopping operation according to the present invention.

Fig. 8B is a flowchart showing a preview stopping operation according to the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Referring now to Fig. 1, there is shown a block diagram for practicing the 25 present invention. In one embodiment, the present invention is implemented on a personal computer running the Microsoft® Windows™ 95 operating system on an Intel® Pentium® processor. The present invention may be implemented as a feature in a software application such as a browser running on the aforementioned hardware.

One skilled in the art will recognize that other embodiments, including other types of software applications, processors, and operating systems, are also possible.

In the block diagram of Fig. 1, computer 100 is shown having a central processing unit (CPU) 101, display device 102, input device such as a pointing device 103, random access memory (RAM) 104, and storage device 105. Storage device 105 also includes a cache 106 for temporary storage of link preview data, as will be described in more detail below. Also provided is a connection 107 to a network such as the Internet. The following detailed description of the invention will make reference to exemplary implementations of such components, though other embodiments may 10 also be used. For example, CPU 101 is a microprocessor such as an Intel Pentium processor; display device 102 is a conventional monitor or screen such as a cathode-ray tube (CRT); pointing device 103 is a mouse or trackball, though other input devices such as keyboards can also be used; RAM 104 is some quantity of conventional memory as is commonly supplied with personal computers; storage 15 device 105 is a hard disk or similar device for long-term storage of programs and data; cache 106 is implemented in some portion of storage device 105, though cache 106 can also be implemented in RAM 104 as is known in the art; and Internet connection 107 is implemented using known protocols such as Transmission Control Protocol / Internet Protocol (TCP/IP) across a modem, T1 or T3 line, or other 20 connection medium. Documents to be viewed and previewed using the present invention can be retrieved across Internet connection 107 using known techniques. Internet connection 107 is not necessary to practice the present invention, as the invention can also be implemented for viewing hypertext documents that are stored locally, such as on storage device 105, for example.

25 A user runs a software application by loading the software onto computer 100 (from storage device 105) and invoking a command to begin execution. Typically, the software application is then loaded into RAM 104 so that CPU 101 can have access to the program instructions contained therein. On some computer systems, certain

portions of the software application may remain on disk 105 to be "swapped into" RAM 104 only as needed, in order to conserve memory space.

One example of such a software application is a browser. Such browsers are well known in the art, including for example Netscape Navigator and Microsoft Internet Explorer. For illustrative purposes, the remainder of this description will focus on browser applications for viewing documents and pages on a network over an Internet connection, though the present invention can easily be implemented in connection with other types of applications and documents as well, such as for example word processing applications and/or locally-stored documents. For purposes of this description, the terms "document" and "page" are used interchangeably, though other types of documents or files could also be used without departing from the spirit or essential characteristics of the invention.

Referring now to Fig. 2, there is shown an example of a hypertext document 200. Document 200 is stored as an HTML file at some location on a remote server, or may be locally stored. As is known in the art, a user may retrieve document 200 using an Internet Protocol (IP) address or Uniform Resource Locator (URL) specifying the location of document 200, and document 200 is then displayed on a screen 102 using browser software.

Document 200 contains text and/or graphics, as shown in the example of Fig. 2. Links 201, 202, 203 to other documents 211, 212, 213 may be included within document 200, and may be represented as graphic elements 201, 202, or as text 203. Links 203 associated with text are typically displayed in a unique color, typeface, and/or style to indicate to the user that the associated text is a link. Links 201, 202 associated with graphic elements may be indicated by a colored border, if desired. Other means of highlighting links 201, 202, 203 may be employed, and such means may be selectable or configurable by the user. In some cases, it may be desirable not to highlight or indicate the presence of links.

For purposes of this application, a document such as document 200 is referred to as a "source" document, whereas those such as documents 211, 212, 213 are referred to as "linked" documents.

As is known in the art, a user may retrieve a linked document 211, 212, 213 by activating the appropriate link 201, 202, 203 corresponding to the desired document. Thus, it is advantageous if link 201, 202, 203 somehow indicates to the user the content of the corresponding document 211, 212, 213. Typically, the author of document 200 provides such indications by describing the linked document's contents in the graphic or text element associated with the link. Additional indications of the nature or content of the linked document may be provided by specifying text in an "ALT" tag in the link specification. In some browsers, such as Microsoft Internet Explorer, such text is displayed momentarily on the screen when the user passes an on-screen cursor over the location of the link. Thus, the text may include further description of the contents of the linked document. In addition, many browsers display the URL or file name of a linked document in a status bar whenever an on-screen cursor is passed over a link.

Referring now to Fig. 3, there is shown an example of a screen display of a hypertext document 301 according to the prior art. Document 301 contains text and graphic images; links 302 are indicated and distinguished from normal text by the use of underlining. On-screen cursor 304 can be moved by the user by manipulating pointing device 103. The user activates a link 302 by clicking a button, such as a mouse button, while cursor 304 is positioned over the desired link 302.

In the example of Fig. 3, the user has positioned cursor 304 over link 303. The user is given some indication of the contents of the linked document by the text associated with the link, namely the words "Hewlett-Packard". In addition, a task bar 305 is provided, which displays the URL of the linked document, namely "<http://www.hp.com>". However, as discussed above, these indications (along with the optional ALT tag, not shown), are insufficient to provide an accurate description of the current contents of the linked document, particularly when such contents may

have changed since the source document 301 was authored. All of the indications shown in Fig. 3 are provided by the author of document 301 only, and are not drawn from or based on the contents of the linked documents themselves. Thus, such indications cannot reflect changes in the linked documents unless the author 5 manually makes corresponding changes to the links in document 301. In addition, none of the indications shown in Fig. 3 provide any graphical representation of the contents of linked documents.

Referring now to Fig. 4, there is shown an example of a screen display of a 10 hypertext document 401 including a link preview 402 according to one embodiment of the present invention. When the user positions cursor 303 over a link 302 and holds cursor 303 in that position for some predetermined length of time, a preview 402 of the linked document is shown. Thus, the user is able to see some portion of the contents of the linked document, and is better able to make a decision as to whether or not to follow the link.

15 In the example of Fig. 4, link preview 402 is shown as a miniature representation of a portion of the linked document, presented in a small window overlapping the display of source document 401. Other arrangements and techniques for displaying link preview 402 may also be used. In addition, the invention may 20 operated in other manners than that described in this example. For example, multiple previews of links could be displayed, each adjacent to the corresponding link. Alternatively, link preview 402 could be displayed only when the user requests such display, for example by clicking a button or activating a command. In yet other embodiments, link previews 402 can be displayed automatically for all or some on-screen links, without requiring the user to express interest in a particular link. Those 25 skilled in the art will recognize that many other implementations of the present invention are possible.

Referring now to Fig. 5, there is shown a block diagram of the functional components of one embodiment of the present invention. In one embodiment, each of the functional components of Fig. 5 is implemented as a software object forming

part of a software application, as described below. In an alternative embodiment, the functional components could be implemented according to some other architecture, as will be recognized by those skilled in the art.

- Link Previewer 501 initiates and coordinates the process of generating and
- 5 displaying link previews 402. Link Previewer 501 performs the following functions:
- receive messages from Page Viewer 502 that indicate when a link 302, 303 is present in a document;
 - send messages to initiate loading of link previews 402 using Resource Provider 504;
 - 10 • respond to user interest or termination of user interest in a link 302, 303 as reported by Page Viewer 502;
 - initiate rendering of link preview 402 using Preview Player 503; and
 - initiate display of link preview 402 in Page Viewer 502 when the user has shown interest in a link 303.
- 15 Page Viewer 502 displays source document 401, and delegates a portion of its graphics area to Link Previewer 501 for preview 402 display. Page Viewer 502 also notifies Link Previewer 501 when certain events take place, such as:

- a new document 401 including links 302, 303 is loaded;
- the user indicates interest in a link 303; and
- 20 • the user terminates interest in a link 303.

Page Viewer 502 may be implemented using a conventional software system for displaying HTML documents, with appropriate non-conventional enhancements for implementing the features and methods of the present invention.

- Preview Player 503 renders a link preview 402 in a portion of Page Viewer's
- 25 502 graphics area. Preview Player 503 may also be utilized in link preview resource loading by determining when enough of a link preview resource has been loaded to display the preview 402. In one embodiment, Preview Player 503 is implemented as a portion of Page Viewer 502.

Resource Provider 504 retrieves link preview resources using a location identifier. A location identifier identifies the contents or target of a link. Link preview resources are generally the contents or target themselves, but may be any form of content. In one embodiment, location identifiers are uniform resource locators (URLs), and link preview resources are linked web pages. One skilled in the art will recognize, however, that any content or application may be provided as a resource, and any form of location identifier may be employed, without departing from the spirit or essential characteristics of the present invention.

Location identifiers are provided for Resource Provider 504, which loads the specified content. In one embodiment, Resource Provider 504 performs such functions in a manner that is transparent to the remaining elements of the system. Resource Provider 504 performs such operations in response to signals from Link Previewer 501; it also stops retrieval of resources in response to signals from Link Previewer 501 indicating that a resource is no longer needed or if enough of it has already been loaded. In one embodiment, Resource Provider 504 is capable of loading several resources simultaneously, as needed, though such functionality is not necessary.

Resource Cache 505, which may be implemented in cache 106 of storage device 105, is a repository for stored link preview resources. When a resource is loaded at the request of Link Previewer 501, it is stored in Cache 505 for later retrieval. Resource Cache 505 may store, for example, portions of linked content identified on a document that is currently loaded. In one embodiment, Page Viewer 502 may also use resources stored in Resource Cache 505 for display of the linked document when the user selects a link and thereby requests retrieval of the full document. Thus, Resource Cache 505 aids rendering a link preview more quickly when the user expresses interest, and may also allow linked content to be pre-loaded to improve performance should the user follow the link.

In one embodiment, Resource Cache 505 includes the following methods:

- cacheResource: Stores a resource or other content for later retrieval.

- `retrieveResource`: Returns stored content indexed by location, specified by a location identifier.

Timer 506 is a conventional timer for notifying Link Previewer 501 that a certain amount of time has passed. This information may be used in one embodiment 5 to stop display of a preview 402 after a predetermined length of time has passed. Such length of time may be configurable by the user. In another embodiment, timer 506 may be used to initiate previewing operations only when the user positions the on-screen cursor over a link for a specified length of time. This may be useful in avoiding unwanted preview operations when the user causes the cursor to pass over 10 a link on the way to pointing to another area of the screen. In such a situation, the user has no interest in the link, so that initiating a preview is not desirable.

In one embodiment, Timer 506 includes the following methods:

- `startTimer(waittime, expirable)`: Causes timer to count down until waittime has elapsed.
- `expire()`: Expires an expirable object passed into `startTimer`.

In another embodiment, timer 506 may be absent, as it is not necessary to practice the present invention.

The present invention operates in one embodiment by loading resources needed for displaying previews 402 for all links 302, 303 as soon as the source 20 document 401 has been loaded. When the user indicates interest in a particular link 302, the corresponding preview 402 can immediately be displayed if its resources have been loaded. If the loading has not yet been performed or completed when a user expresses interest in a link 303, Link Previewer 501 issues a signal to initiate immediate loading of the resources needed for that link 303. When such loading is 25 complete, the preview 402 is displayed for the user. Preview 402 may be stopped when appropriate, for example when a predefined period of time has elapsed, or when the user indicates that he or she is no longer interested in the link 302.

Referring now to Figs. 6A and 6B, there are shown a timeline diagram and flowchart of a document loading operation according to one embodiment of the

present invention. Document 401 is loaded in response to a request from the user wherein he or she specifies the URL or IP address and location of the desired document. This process is initiated by sending 601 a loadPage(location) instruction to Page Viewer 502. For each link in document 401, steps 602 through 606 are

5 performed. First, Page Viewer 502 notifies 602 Link Previewer 501 of the link using a seenLink(location) call. The seenLink(location) call passes the location identifier to Link Previewer 501 to initiate the preview process, and effectively acts as an entry point for caching linked content for previewing. Link Previewer 501 calls Resource Cache 505 with an isLinkLoaded(location) call to determine 603 whether or not the

10 link has already been loaded. If the link has not been loaded, and if in 604 it is determined that loading of the link has not already begun, Link Previewer 501 calls Resource Provider 504 with a loadLink(location) command to initiate loading. Resource Provider 504 initiates loading, and sends a cacheResource(location, resource) command 606 to Resource Cache 505 in order to cache to loaded link for

15 future use.

Operations shown in Figs. 6A and 6B may proceed in parallel, so that some links may continue to load while other operations take place. Thus, loading may still be in process when the user expresses an interest in a link, as described below in connection with Figs. 7A and 7B.

20 Referring now to Figs. 7A and 7B, there are shown a timeline diagram and flowchart of a link previewing operation according to one embodiment of the present invention. The method of Figs. 7A and 7B allow for concurrent loading operations as described above, with priority given to any link in which the user expresses interest. In other words, the system according to one embodiment forces immediate loading of

25 preview data for a link whenever the user holds the cursor over that link or otherwise expresses interest.

A user expresses interest 700 in a link generally by positioning an on-screen cursor over the link and holding that position for some predetermined length of time, such as for example one-half second, though any length of time may be used.

Typically, Page Viewer 502 receives a signal indicating that the cursor has been so positioned, as is known in the art for the display of ToolTips or similar descriptive information concerning the link.

Page Viewer 502 sends an interest(location) command 701 to Link Previewer 501, indicating that a preview should be displayed for the specified location (i.e., link). Link Previewer 501 then sends an isLinkLoaded(location) command 702 to Resource Cache 505 to determine whether preview data for the specified link has already been loaded. If not, Link Previewer 501 sends a forceLoad(location) command 703 to Resource Provider 504 forcing immediate loading of the relevant 10 preview data. This ensures that priority will be given to the display of the particular link the user has indicated, without wasting undue time loading resources for other links. Loading of other links can resume when it is possible to do so without compromising the speed or efficiency of loading the desired link.

Link Previewer 501 sends a getPreviewArea(size) command 704 to Page Viewer 502. The getPreviewArea(size) command provides a GraphicsArea parameter to Preview Player 503, so that Preview Player 503 can display the preview in the appropriate location. Link Previewer 501 then sends a playPreview(size, GraphicsArea) command 705 to Preview Player 503, which begins to display the preview in the specified location on the display screen. This display may show a 20 miniature representation of a portion of the linked document, as shown in the example of Fig. 1.

In one embodiment, Link Previewer 501 also sends a startTimer(duration) command 706 to Timer 506, specifying a duration for display of the preview. The purpose of this step is to provide a mechanism for stopping the display of the 25 preview after some predetermined length of time, so that the preview is not obtrusive to the user. This length of time may be user-configurable, and there may also be alternative mechanisms for the user to request that the preview display be stopped. In addition, if the user moves the on-screen cursor so that it is no longer positioned

over the link being previewed, the preview is stopped, as will be described below in connection with Figs. 8A and 8B.

Once Timer 506 determines that the duration of time has expired, it sends an expired() command 707 to Link Previewer 501, which in turn sends a stopPreview() command 708 to Preview Player 503. Preview Player 503 stops displaying the preview. Link Previewer 501 also sends a refresh() command 709 to Page Viewer 502 which refreshes the area of the display that was occupied by the preview.

Referring now to Figs. 8A and 8B, there are shown a timeline diagram and flowchart of a preview stopping operation according to one embodiment of the 10 present invention. These Figures show operation of the present invention when the user indicates that he or she is no longer interested in a link.

The user indicates 800 that he or she is no longer interested in a link, typically by moving the on-screen cursor so that it is no longer positioned over the link. Page Viewer 502 detects this, and sends a notInterested(location) command 801 to Link 15 Previewer 501 to signal the event. Link Previewer 501 sends stop(location) commands 802, 803, 804, 805 to Preview Player 503, Resource Provider 504, Resource Cache 505, and Timer 506, respectively. This serves to cease display of the link preview by notifying all components that the user is no longer interested in the link.

In one embodiment, loading of link preview resources continues concurrently 20 with the steps of Figs 7A, 7B, 8A, and 8B. Thus, previews of linked documents are continuously loaded until loading is complete, so as to minimize display of a preview when the user expresses interest in a link.

From the above description, it will be apparent that the invention disclosed herein provides a novel and advantageous system and method of displaying a 25 preview of a linked document before the link is activated. The foregoing discussion discloses and describes merely exemplary methods and embodiments of the present invention. As will be understood by those familiar with the art, the invention may be embodied in other specific forms without departing from the spirit or essential characteristics thereof. Accordingly, the disclosure of the present invention is

intended to be illustrative, but not limiting, of the scope of the invention, which is set forth in the following claims.

We Claim:

1. In a system for displaying hypertext documents, a computer-implemented method of displaying a preview of a linked document, comprising:
 - a) retrieving a source document containing at least one link to a linked document;
 - b) receiving a selection identifying one of the links;
 - c) retrieving preview data representing at least a portion of the linked document corresponding to the identified link; and
 - d) displaying the retrieved preview data.
- 10 2. The computer-implemented method of claim 1, wherein c) comprises:
 - c.1) retrieving at least a portion of the linked document; and
 - c.2) generating preview data from the retrieved portion of the linked document.
3. The computer-implemented method of claim 1, further comprising:
 - 15 e) stopping display of the retrieved preview data after a predetermined time period.
4. The computer-implemented method of claim 1, wherein the at least one link has an on-screen position, and wherein b) comprises detecting that a position of an on-screen cursor corresponds to the position of the link.
- 20 5. The computer-implemented method of claim 4, further comprising:
 - e) stopping display of the retrieved preview data responsive to the position of the on-screen cursor no longer corresponding to the position of the link.

6. The computer-implemented method of claim 1, further comprising:
- e) stopping display of the retrieved preview data responsive to the position of the on-screen cursor no longer corresponding to the position of the link.

5 7. The computer-implemented method of claim 1, further comprising: subsequent to step a), for each link in the source document:

- a.1) retrieving preview data representing at least a portion of the linked document corresponding to the link; and
- a.2) caching the retrieved preview data in a storage cache;

10 and wherein step c) comprises retrieving preview data from the storage cache.

8. The computer-implemented method of claim 1, further comprising: subsequent to step a), for each link in the source document:

- a.1) commencing retrieval of preview data representing at least a portion of the linked document corresponding to the link; and
- a.2) commencing caching of the retrieved preview data in a storage cache;

15

and wherein step c) comprises:

- c.1) responsive to retrieval and caching of the preview data corresponding to the identified link being complete, retrieving preview data from the storage cache; and
- c.2) responsive to retrieval and caching of the preview data corresponding to the identified link not being complete, forcing retrieval of preview data corresponding to the identified link.

20

9. The computer-implemented method of claim 1, wherein d) comprises displaying the retrieved preview data in an overlay window adjacent to the identified link.

10. The computer-implemented method of claim 1, wherein d) comprises 5 displaying a reduced-size representation of a portion of the linked document in an overlay window adjacent to the identified link.

11. A system for displaying a preview of a linked document in a hypertext document display application, comprising:

10 a retrieval module for retrieving a source document containing at least one link to a linked document;

an input device coupled to the retrieval module, for receiving a selection identifying one of the links;

15 a preview generating module, coupled to the retrieval module, for retrieving preview data representing at least a portion of the linked document corresponding to the identified link; and

a viewer, coupled to the preview generating module, for displaying the retrieved preview data.

12. The system of claim 11, wherein the preview generating module comprises:

20 a resource provider for retrieving at least a portion of the linked document; and

a link previewer, coupled to the resource provider, for generating preview data from the retrieved portion of the linked document.

13. The system of claim 11, wherein the at least one link has an on-screen 25 position, and wherein the input device detects that a position of an on-screen cursor corresponds to the position of the link.

14. The system of claim 13, wherein the preview generating module stops display of the retrieved preview data responsive to the position of the on-screen cursor no longer corresponding to the position of the link.

15. The system of claim 11, further comprising:
5 a resource cache coupled to the retrieval module for, for each link in the source document, retrieving preview data representing at least a portion of the linked document corresponding to the link, and caching the retrieved preview data in a storage cache;

and wherein the preview generating module retrieves preview data from the
10 storage cache.

16. The system of claim 11, wherein the viewer displays the retrieved preview data in an overlay window adjacent to the identified link.

17. The system of claim 11, wherein the viewer displays a reduced-size representation of a portion of the linked document in an overlay window
15 adjacent to the identified link.

18. A computer program product comprising a computer-readable medium having computer readable code embodied therein for displaying a preview of a linked document in a system for displaying hypertext documents, the computer program product comprising:
20 computer-readable program code devices configured to cause a computer to retrieve a source document containing at least one link to a linked document;
computer-readable program code devices configured to cause a computer to receive a selection identifying one of the links;

computer-readable program code devices configured to cause a computer to
retrieve preview data representing at least a portion of the linked
document corresponding to the identified link; and
computer-readable program code devices configured to cause a computer to
5 display the retrieved preview data.

19. The computer program product of claim 18, wherein the computer-
readable program code devices configured to cause a computer to retrieve preview
data comprise:

computer-readable program code devices configured to cause a computer to
10 retrieve at least a portion of the linked document; and
computer-readable program code devices configured to cause a computer to
generate preview data from the retrieved portion of the linked
document.

20. The computer program product of claim 18, further comprising:

15 computer-readable program code devices configured to cause a computer to
stop display of the retrieved preview data after a predetermined
time period.

21. The computer program product of claim 18, wherein the at least one link
has an on-screen position, and wherein the computer-readable program code devices
20 configured to cause a computer to receive a selection identifying one of the links
comprise computer-readable program code devices configured to cause a computer
to detect that a position of an on-screen cursor corresponds to the position of the link.

22. The computer program product of claim 21, further comprising:

computer-readable program code devices configured to cause a computer to
25 stop display of the retrieved preview data responsive to the position

of the on-screen cursor no longer corresponding to the position of the link.

23. The computer program product of claim 18, further comprising: computer-readable program code devices configured to cause a computer to stop display of the retrieved preview data responsive to the position of the on-screen cursor no longer corresponding to the position of the link.

5

24. The computer program product of claim 18, further comprising: computer-readable program code devices configured to cause a computer to, for each link in the source document: retrieve preview data representing at least a portion of the linked document corresponding to the link; and cache the retrieved preview data in a storage cache;

10

and wherein the computer-readable program code devices configured to cause a computer to retrieve preview data comprise computer-readable program code devices configured to cause a computer to retrieve preview data from the storage cache.

15

25. The computer program product of claim 18, further comprising: computer-readable program code devices configured to cause a computer to, for each link in the source document: commence retrieval of preview data representing at least a portion of the linked document corresponding to the link; and commence caching of the retrieved preview data in a storage cache;

20

and wherein the computer-readable program code devices configured to cause a computer to retrieve preview data comprise:

25

computer-readable program code devices configured to cause a computer to, responsive to retrieval and caching of the preview data corresponding to the identified link being complete, retrieve preview data from the storage cache; and

5 computer-readable program code devices configured to cause a computer to, responsive to retrieval and caching of the preview data corresponding to the identified link not being complete, force retrieval of preview data corresponding to the identified link.

26. The computer program product of claim 18, wherein the computer-
10 readable program code devices configured to cause a computer to display the retrieved preview data comprise computer-readable program code devices configured to cause a computer to display the retrieved preview data in an overlay window adjacent to the identified link.

27. The computer-implemented method of claim 18, wherein the computer-
15 readable program code devices configured to cause a computer to display the retrieved preview data comprise computer-readable program code devices configured to cause a computer to display a reduced-size representation of a portion of the linked document in an overlay window adjacent to the identified link.

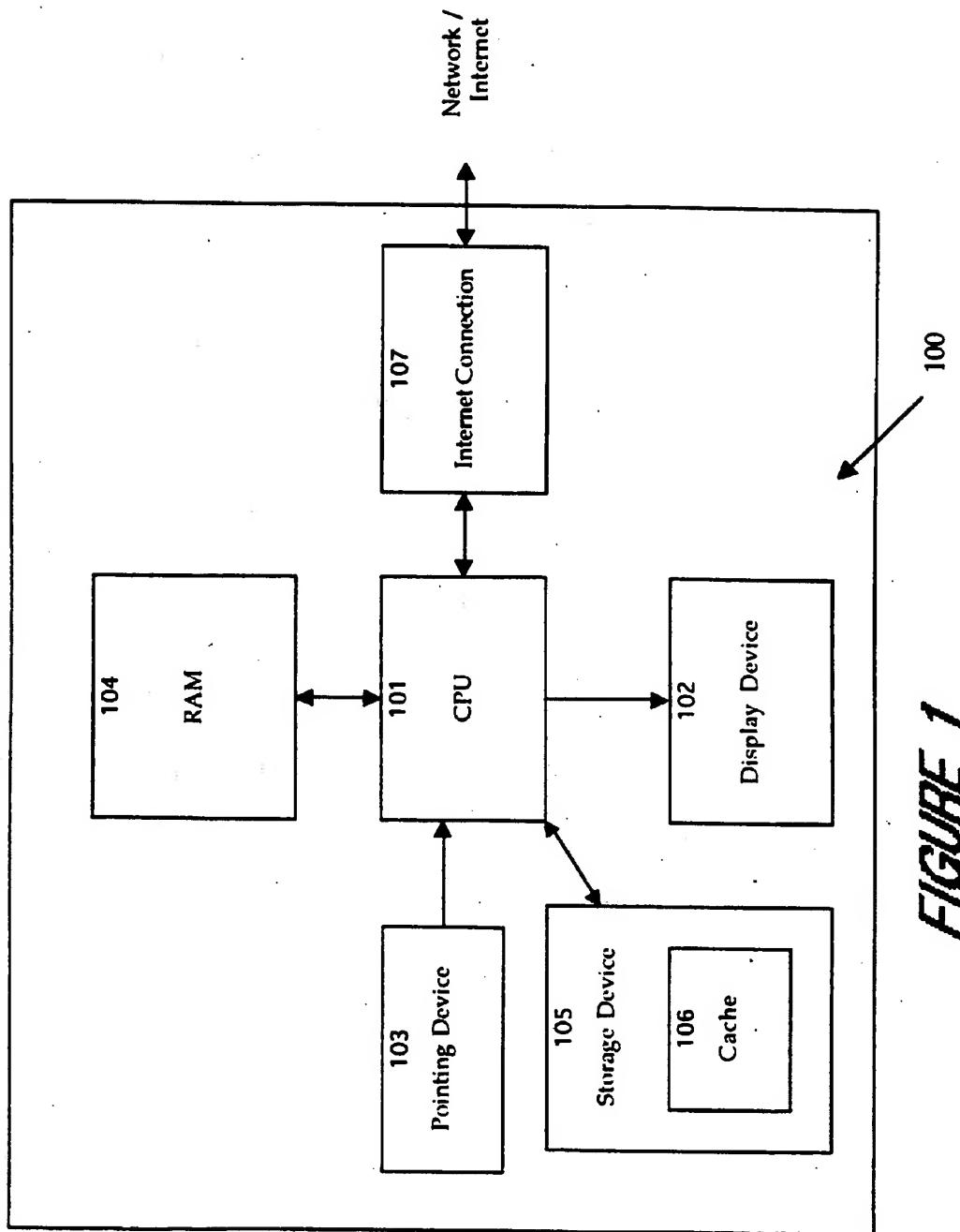
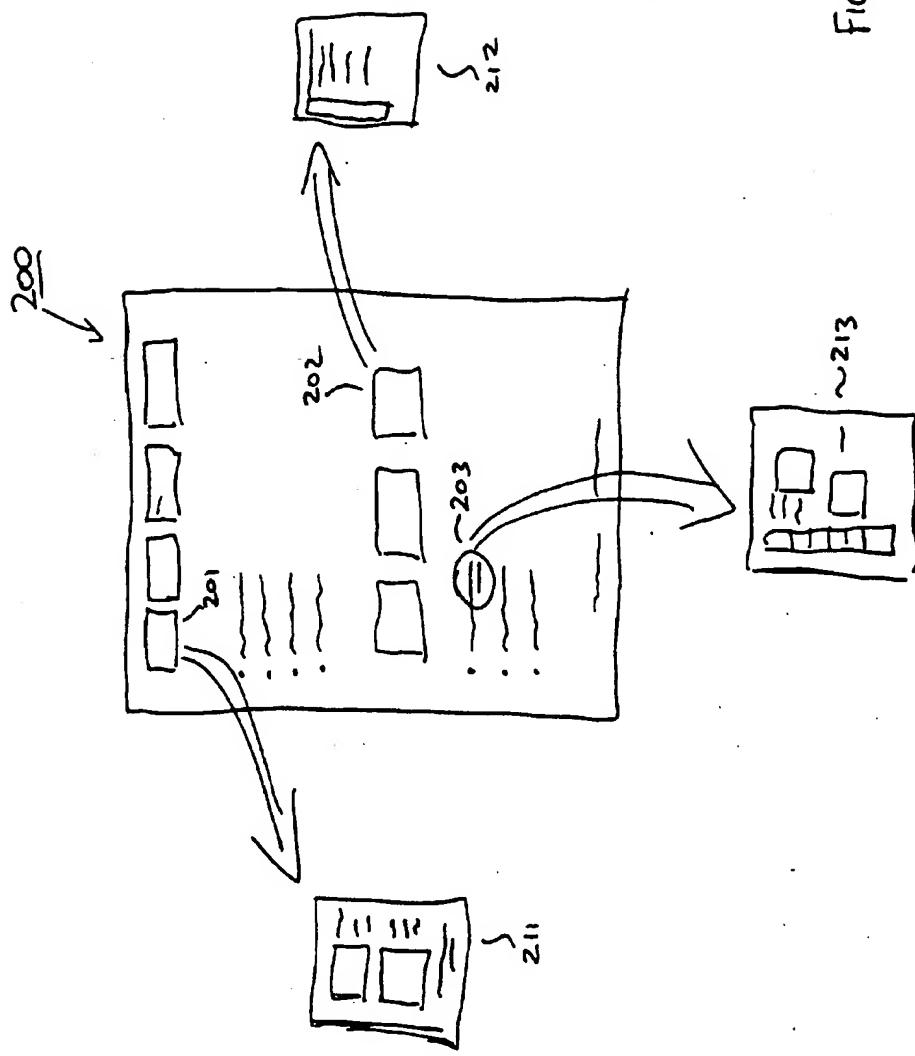


FIGURE 1

FIG 2



File Edit View Go Bookmarks Options Directory Window Connected

Back Home Edit Reload Images Print Find Stop

NetSite: <http://www.corel.com/partners/partners.htm>

What's New? What's Cool? Destinations Net Search People Software

COREL Home Products Support Shop Contact Us News Events Search

Corel Partners

OEM Partners

- Epson ~ 302
- Hewlett-Packard ~ 303
- Packard Bell
- Ubiquity ~ 302
- Vobis ~ 302

Technology Partners

- Dragon Systems, Inc.
- AmpliCode Software
- Bluefish International Pte. Ltd.
- Borland
- CNTC
- Hoffmann + Associates
- INSO
- KMT Software
- Netscape
- Novell
- Starfish Software
- Software Development Corporation
- Sgento
- TriTech Corporation

Learning Partners

- Approved Curriculum Vendors
- Conferences and Seminars
- Magazines, Books, Newsletters
- SchoolNet Database

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

8010

8011

8012

8013

8014

8015

8016

8017

8018

8019

8020

8021

8022

8023

8024

8025

8026

8027

8028

8029

8030

8031

8032

8033

8034

8035

8036

8037

8038

8039

8040

8041

8042

8043

8044

8045

8046

8047

8048

8049

8050

8051

8052

8053

8054

8055

8056

8057

8058

8059

8060

8061

8062

8063

8064

8065

8066

8067

8068

8069

8070

8071

8072

8073

8074

8075

8076

8077

8078

8079

8080

8081

8082

8083

8084

8085

8086

8087

8088

8089

8090

8091

8092

8093

8094

8095

8096

8097

8098

8099

80100

80101

80102

80103

80104

80105

80106

80107

80108

80109

80110

80111

80112

80113

80114

80115

80116

80117

80118

80119

80120

80121

80122

80123

80124

80125

80126

80127

80128

80129

80130

80131

80132

80133

80134

80135

80136

80137

80138

80139

80140

80141

80142

80143

80144

80145

80146

80147

80148

80149

80150

80151

80152

80153

80154

80155

80156

80157

80158

80159

80160

80161

80162

80163

80164

80165

80166

80167

80168

80169

80170

80171

80172

80173

80174

80175

80176

80177

80178

80179

80180

80181

80182

80183

80184

80185

80186

80187

80188

80189

80190

80191

80192

80193

80194

80195

80196

80197

80198

80199

80200

80201

80202

80203

80204

80205

80206

80207

80208

80209

80210

80211

80212

80213

80214

80215

80216

80217

80218

80219

80220

80221

80222

80223

80224

80225

80226

80227

80228

80229

80230

80231

80232

80233

80234

80235

80236

80237

80238

80239

80240

80241

80242

80243

80244

80245

80246

80247

80248

80249

80250

80251

80252

80253

80254

80255

80256

80257

80258

80259

80260

80261

80262

80263

80264

80265

80266

80267

80268

80269

80270

80271

80272

80273

80274

80275

80276

80277

80278

80279

80280

80281

80282

80283

80284

80285

80286

80287

80288

80289

80290

80291

80292

80293

80294

80295

80296

80297

80298

80299

80300

80301

80302

80303

80304

80305

80306

80307

80308

80309

80310

80311

80312

80313

80314

80315

80316

80317

80318

80319

80320

80321

80322

80323

80324

80325

80326

80327

80328

80329

80330

80331

80332

80333

80334

80335

80336

80337

80338

80339

80340

80341

80342

80343

80344

80345

80346

80347

80348

80349

80350

80351

80352

80353

80354

80355

80356

80357

80358

80359

80360

80361

80362

80363

80364

80365

80366

80367

80368

80369

80370

80371

80372

80373

80374

80375

80376

80377

80378

80379

80380

80381

80382

80383

80384

80385

80386

80387

80388

80389

80390

80391

80392

80393

80394

80395

80396

80397

80398

80399

80400

80401

80402

80403

80404

80405

80406

80407

80408

80409

80410

80411

80412

80413

80414

80415

80416

80417

80418

80419

80420

80421

80422

80423

80424

80425

80426

80427

80428

80429

80430

80431

80432

80433

80434

80435

80436

80437

80438

80439

80440

80441

80442

80443

80444

80445

80446

80447

80448

80449

80450

80451

80452

80453

80454

80455

80456

80457

80458

80459

80460

80461

80462

80463

80464

80465

80466

80467

80468

80469

80470

80471

80472

80473

80474

80475

80476

80477

80478

80479

80480

80481

80482

80483

80484

80485

80486

80487

80488

80489

80490

80491

80492

80493

80494

80495

80496

80497

80498

80499

80500

80501

80502

80503

80504

80505

80506

80507

80508

80509

80510

80511

80512

80513

80514

80515

80516

80517

80518

80519

80520

80521

80522

80523

80524

80525

80526

80527

80528

80529

80530

80531

80532

80533

80534

80535

80536

80537

80538

80539

80540

80541

80542

80543

80544

80545

80546

80547

80548

80549

80550

80551

80552

80553

80554

80555

80556

80557

80558

80559

80560

80561

80562

80563

80564

80565

80566

80567

80568

80569

80570

80571

80572

80573

80574

80575

80576

80577

80578

80579

80580

80581

80582

80583

80584

80585

80586

80587

80588

80589

80590

80591

80592

80593

80594

80595

80596

80597

80598

80599

80600

80601

80602

80603

80604

80605

80606

80607

80608

80609

80610

80611

80612

80613

80614

80615

80616

80617

80618

80619

80620

80621

80622

80623

80624

80625

80626

80627

80628

80629

80630

80631

80632

80633

80634

80635

80636

80637

80638

80639

80640

80641

80642

80643

80644

80645

80646

80647

80648

80649

80650

80651

80652

80653

80654

80655

80656

80657

80658

80659

80660

80661

80662

80663

80664

80665

80666

80667

80668

80669

80670

80671

80672

80673

80674

80675

80676

80677

80678

80679

80680

80681

80682

80683

80684

80685

80686

80687

80688

80689

80690

80691

80692

806

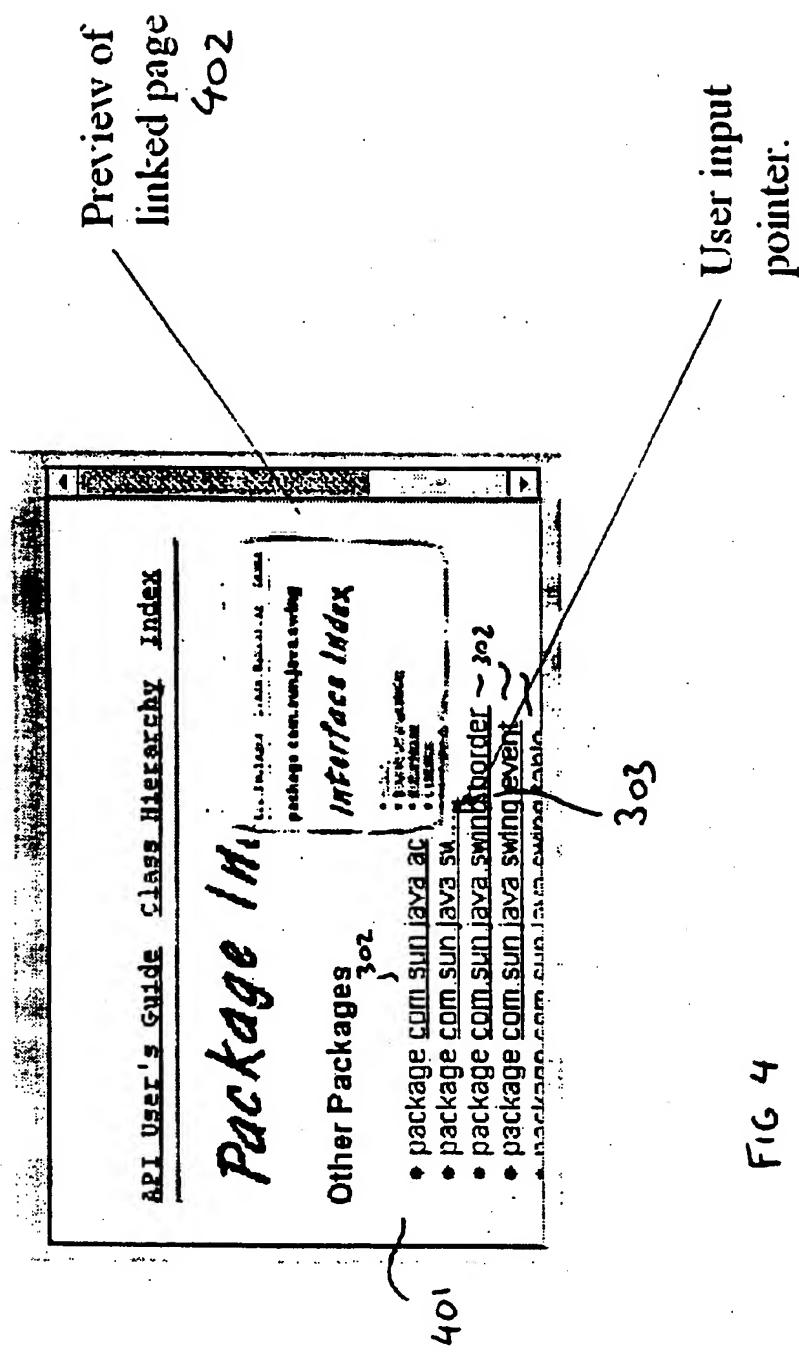


FIG 4

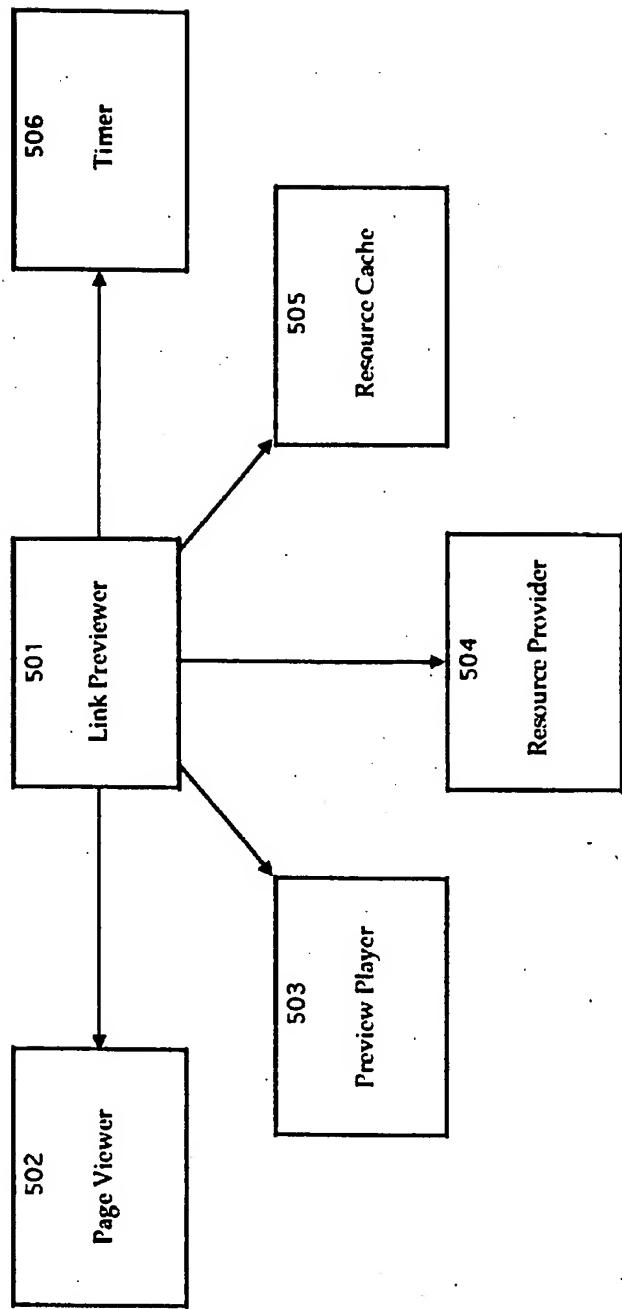


FIGURE 5

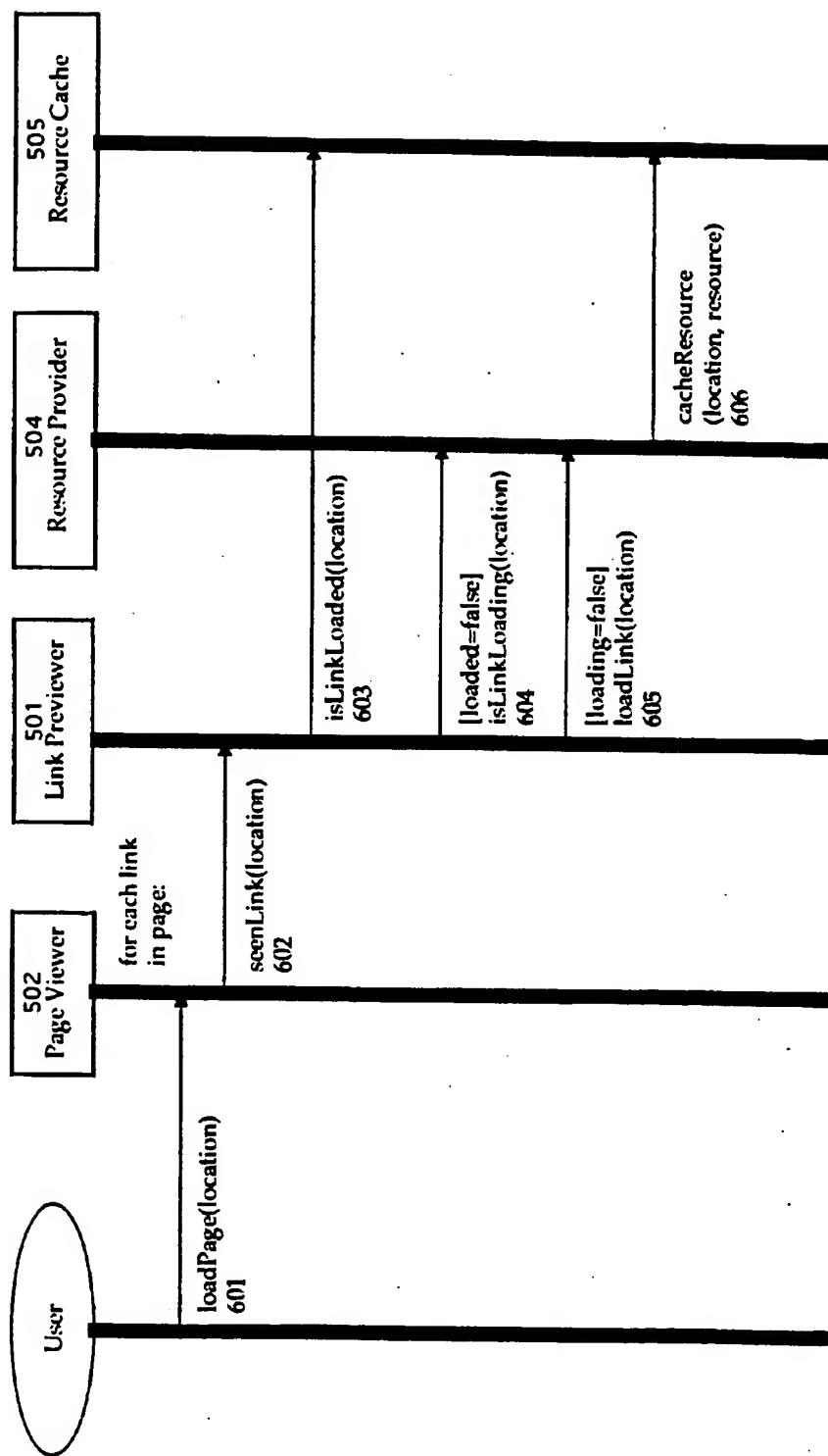
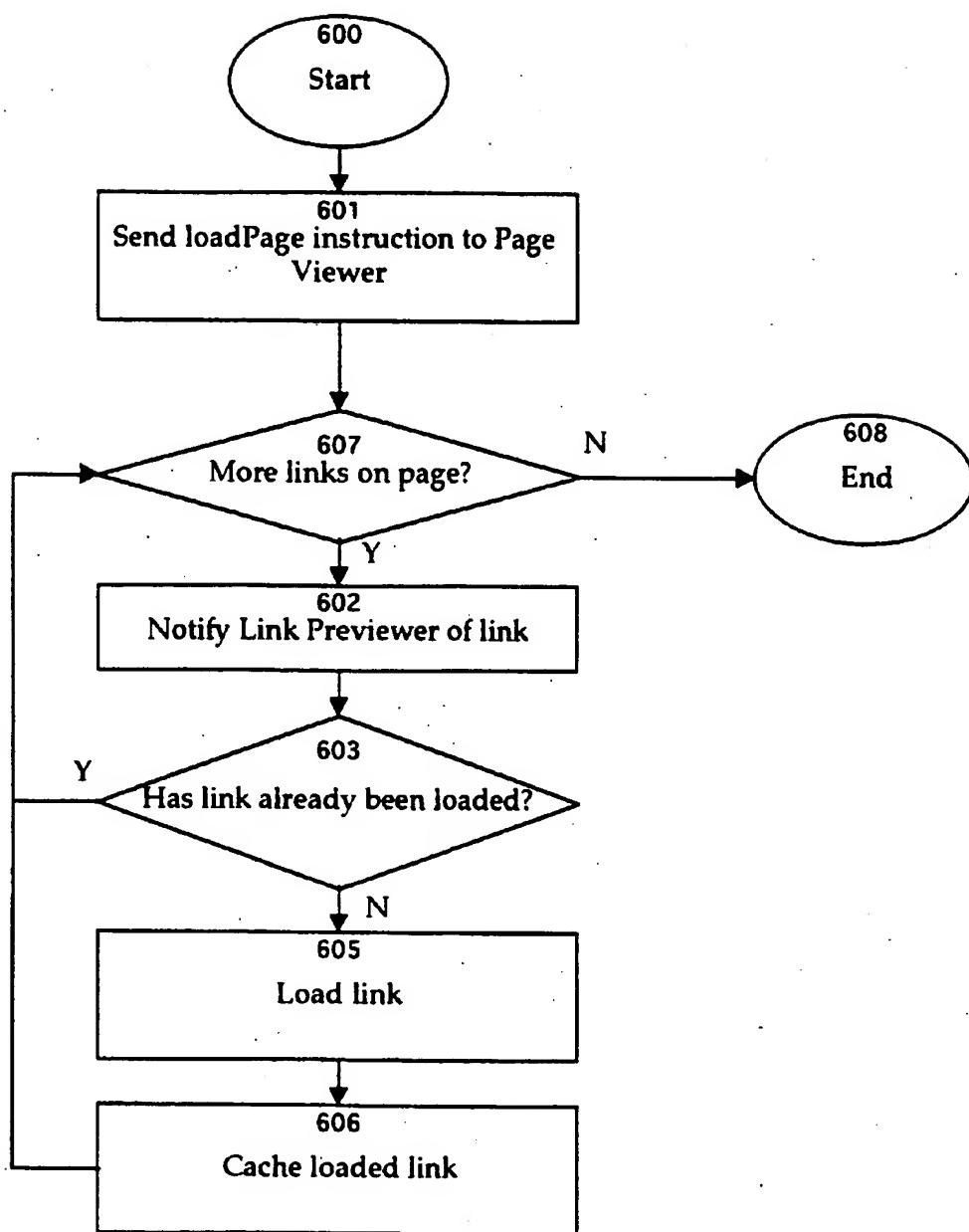


FIGURE 6A

**FIGURE 6B**

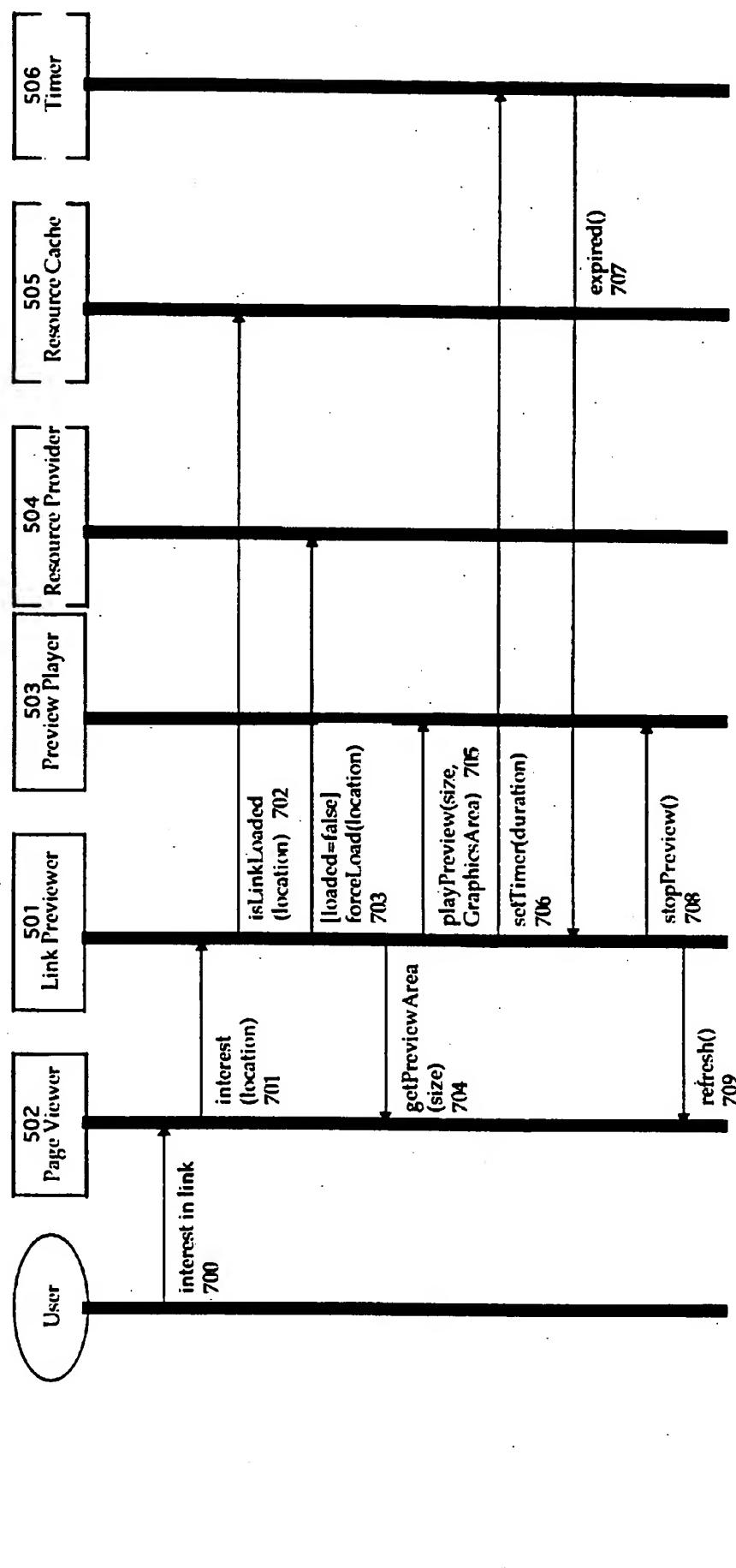
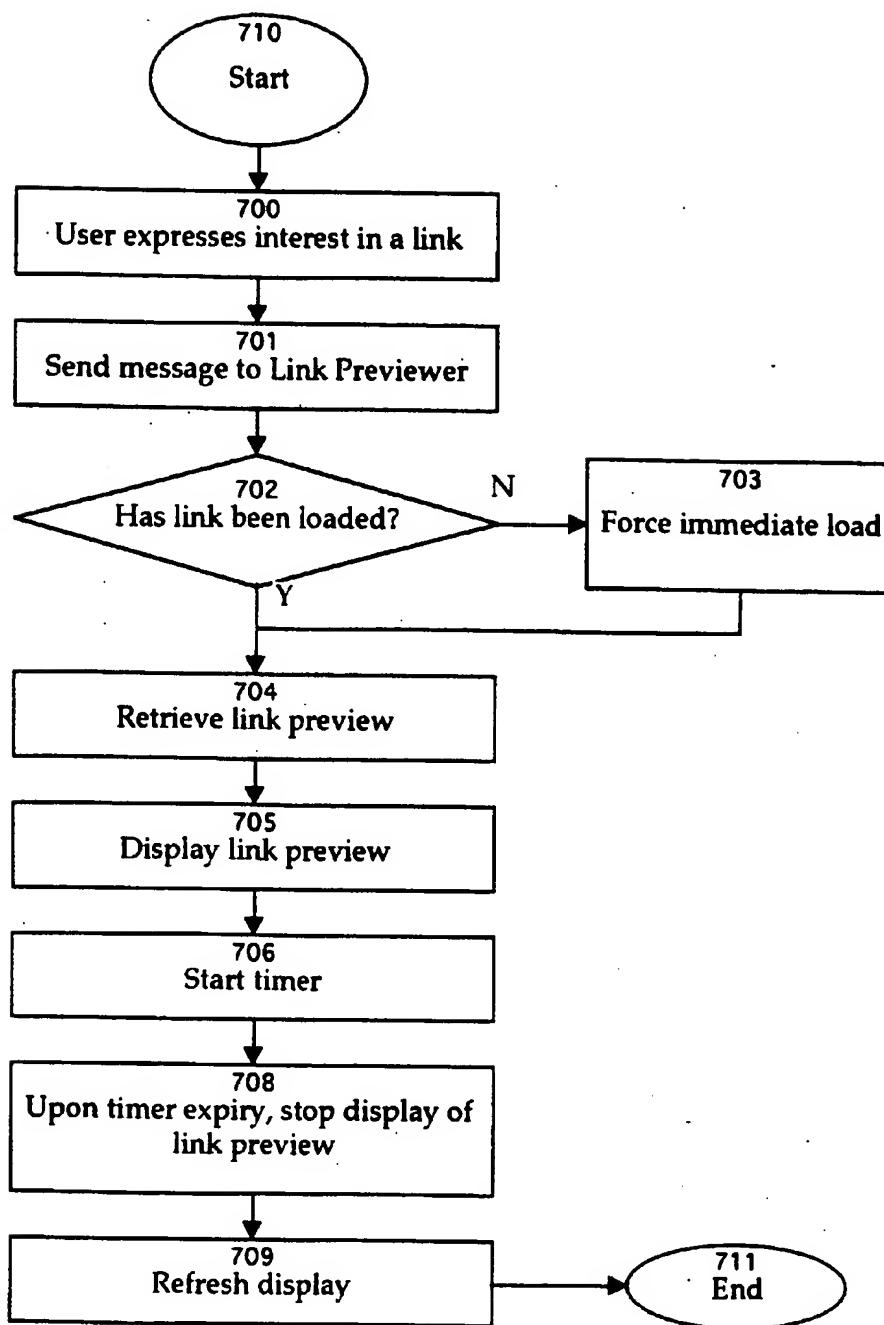


FIGURE 7A

**FIGURE 7B**

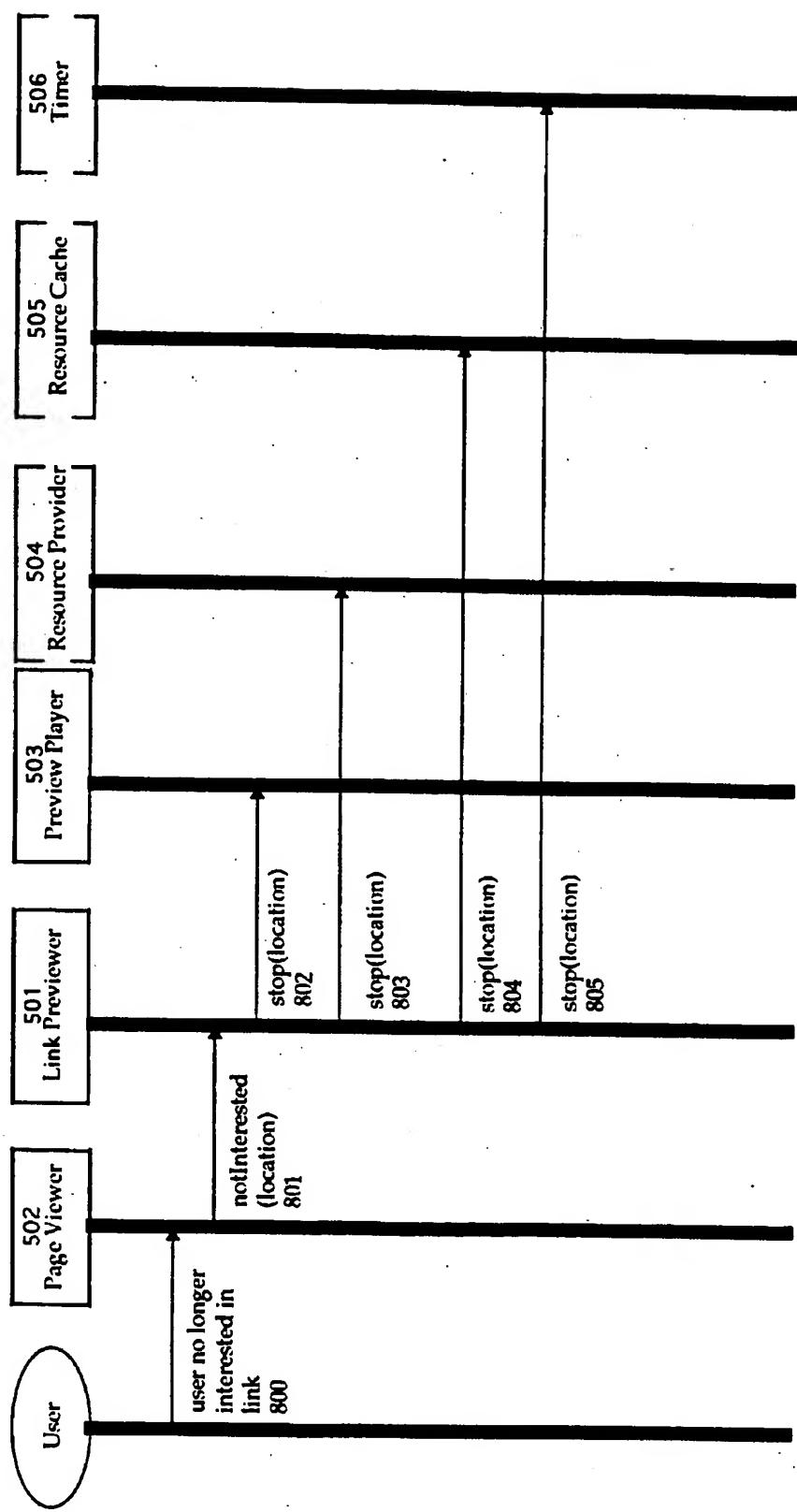
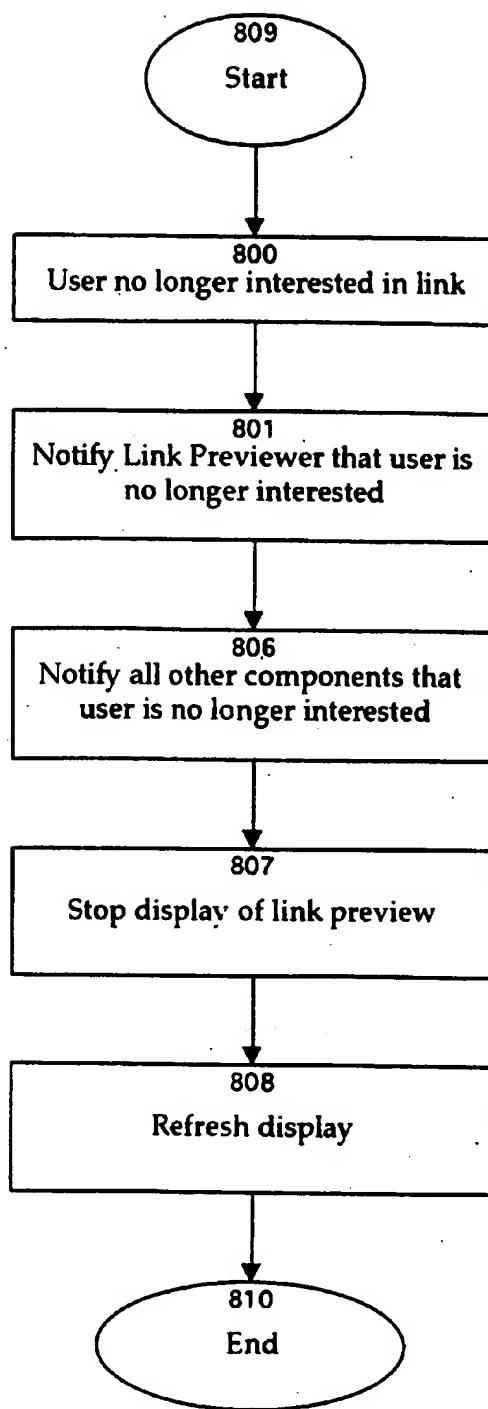


FIGURE 8A

**FIGURE 8B**

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.